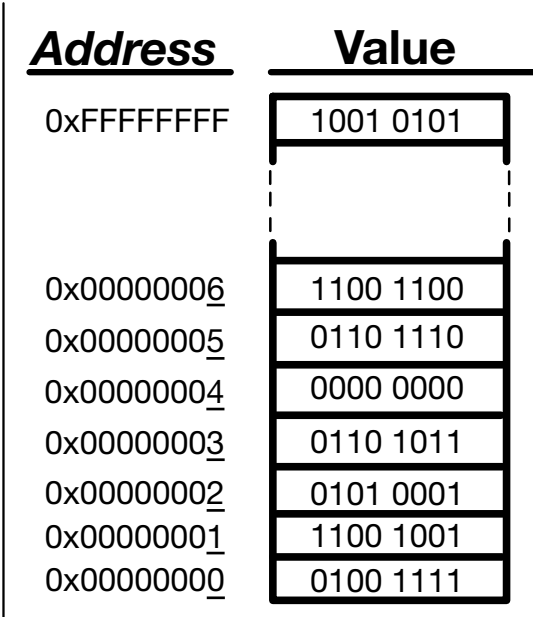
Addresses

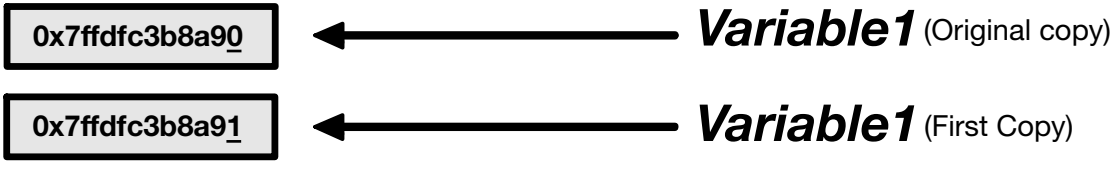
Address- A location in memory where data is stored.

Binary- Base-2: '0' and '1'.
Each digit is called a "Bit".
Computers run off of this.

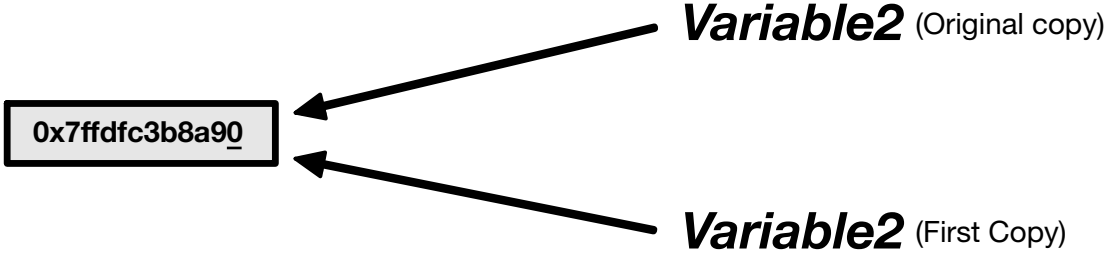
Hexadecimal- Base-16:
Digits: 0 - 9
Letters: A - F
(10 - 15)
Used as addresses for compactness.



Pass by Value



Pass by Reference



```cpp
#include <iostream>

using namespace std;


void addTen(int); //Specifying passing an integer




int main() {
    int variable1 = 0; //Original copy of "variable1" is declared

    cout << "Input a Number: "; cin >> variable1;
    cout << endl << endl << "variable1: " << variable1 << endl;
    addTen(variable1);
    cout << "variable1 Passed: " << variable1 << endl;


    return 0;
}



void addTen(int variable1) { //New copy of "variable1"; Notice its being redeclared in this scope
    /*
    This "variable1" is different than "variable1" in the "main()" function.
    This "variable1" is not in scope of the "main()" function.
    */
    variable1 += 10;
}
```

```
samevans@DESKTOP-ELS67E4: /mnt/c/projects/CS1-SamEvans02/Extra_Material/Passing_Value_vs_Reference/Pass_by_Value
samevans@DESKTOP-ELS67E4:/mnt/c/projects/CS1-SamEvans02/Extra_Material/Passing_Value_vs_Reference/Pass_by_Value$ make && make run
g++ -c -g -Wall -std=c++17 Pass_by_Value.cpp
g++ -o Pass_by_Value.out *.o
./Pass_by_Value.out
Input a Number: 5


variable1: 5
variable1 Passed: 5
samevans@DESKTOP-ELS67E4:/mnt/c/projects/CS1-SamEvans02/Extra_Material/Passing_Value_vs_Reference/Pass_by_Value$
```

```cpp
#include <iostream>

using namespace std;


void addTen(int&); //Specifying passing an address of an integer




int main() {
    int variable1 = 0; //Original copy of "variable1" is declared

    cout << "Input a Number: "; cin >> variable1;
    cout << endl << endl << "variable1: " << variable1 << endl;
    addTen(variable1);
    cout << "variable1 Passed: " << variable1 << endl;


    return 0;
}



void addTen(int& variable1) { //The same copy of "variable1"; Notice the address is what's being passed
    /*
    This "variable1" is the same as "variable1" in the "main()" function.
    This "variable1", while not in scope of the "main()" function, has its address being passed.
    */
    variable1 += 10;
}
```

```
samevans@DESKTOP-ELS67E4: /mnt/c/projects/CS1-SamEvans02/Extra_Material/Passing_Value_vs_Reference/Pass_by_Reference
samevans@DESKTOP-ELS67E4:/mnt/c/projects/CS1-SamEvans02/Extra_Material/Passing_Value_vs_Reference/Pass_by_Reference$ make && make run
g++ -c -g -Wall -std=c++17 Pass_by_Reference.cpp
g++ -o Pass_by_Reference.out *.o
./Pass_by_Reference.out
Input a Number: 5


variable1: 5
variable1 Passed: 15
samevans@DESKTOP-ELS67E4:/mnt/c/projects/CS1-SamEvans02/Extra_Material/Passing_Value_vs_Reference/Pass_by_Reference$
```